Self Studies on Digital Guitar Plug-ins Simulation

Zhixiang Wu

Abstract

This study is about the explanation of the working principles of shelving filter, IIR delay, flanger, and chorus, and the resulting effect they have on my music tracks.

1 Shelving filter

1.1 Introduction and Theory

The track of percussions, bass, violin, and brass are applied with digital filters, which I build in Matlab. The idea is to use shelving filters to enhance the specific frequency range of different tracks, to make the timbre rich, such as increasing the frequency response of specific high frequencies to make the brass and violin brighter. And increasing low-frequency performance for the bass and kick.

The implementation of a shelving filter by Matlab is setting the center frequency of the filter and the gain of the amplitude. By using the mathematical method given by Valimaki and Reiss, a first-order shelving filter could be represented as the transfer function of the input signal and output signal.

$$H_{LS}\ (z) = \frac{b_0' + b_1' z^{-1}}{a_0' + a_1' z^{-1}} \quad (1)$$

$$b_0' = G\tan(\frac{\omega_c}{2}) + \sqrt{G} \quad (2)$$

$$b_1' = G\tan(\frac{\omega_c}{2}) - \sqrt{G} \quad (3)$$

$$a_0' = \tan(\frac{\omega_c}{2}) + \sqrt{G} \quad (4)$$

$$a_1' = \tan(\frac{\omega_c}{2}) - \sqrt{G} \quad (5)$$

Amplitude response of low shelving filter, take inverse of G to make it a high shelving filter. Where ω_c is the center angular frequency, and G is the gain factor. a ,b are the coefficients of the shelving filters.

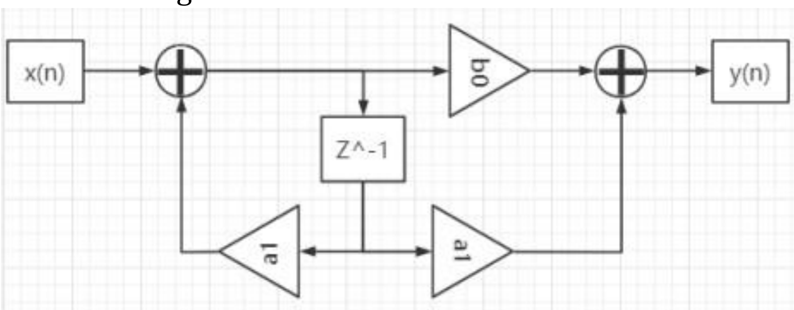

Figure 1: *Structure of Shelving filter*

1.2code demo

```
%% High Shelf
% Gain factor
GH = 4;
% coefficient of LS:
b 0 H = 1/GH*tan(omega c H/2)+sqrt(1/GH);
a 0 H = tan(omega c H/2)+sqrt(1/GH);
b 1 H = 1/GH*tan(omega c H/2)-sqrt(1/GH);
a 1 H = tan(omega c H/2)-sqrt(1/GH);
b0 H = b 0 H/a 0 H;
b1 H = b 1 H/a 0 H;
a1 H = a 1 H/a 0 H;

% CHECK OUT THE MAGNITUDE RESPONSE.

MH= abs((b0 H+b1 H.*Z)./(1+a 1 H.*Z));
% plot(F n,ML);
b0 = GH;
b H = b0*[b0 H b1 H];
a H = [1 a1 H];
[h H,w H] = freqz(b H,a H,fs/2);
semilogx(F_n*fs,mag2db(abs(h_H)));
grid on
```

```
xlabel('Frequency (Hz)')
ylabel('Magnitude (dB)')
y_hp = filter(b H,a H,y);
play_hp = audioplayer(y_hp,fs);
```

1.3 Cascading

Since the timbre of a single instrument has more than one harmonic peak, multiple shelving filters should be applied to a single musical instrument. By adjusting the center frequencies of each peak and notch, and the gain factor for enhancing the amplitude, a simple equalizer is built by cascading those shelving filters together.

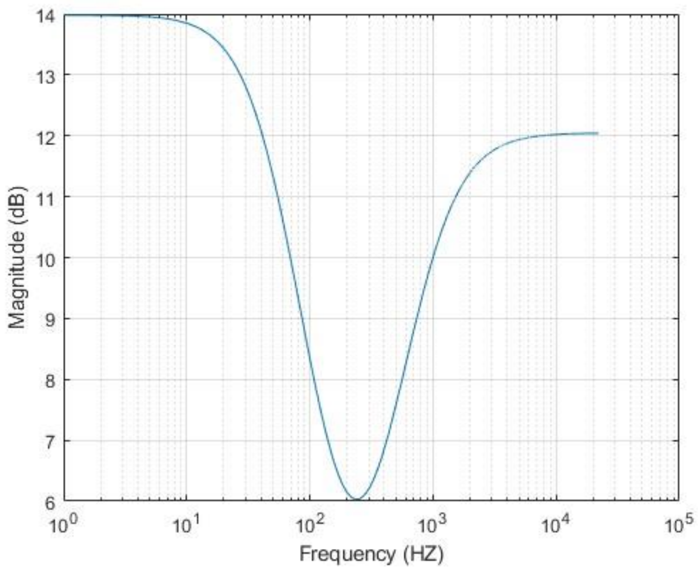

Figure 2: *An example of a second-order shelving filter*

Meanwhile, in order to enrich the timbre further, one extra resonator is implemented by building shelving filters. This is because human auditory has a different sensitivity to different frequencies. This means that the loudness of a sound with a specific sound pressure level differs on different frequencies. The most sensitive frequency of the human hearing system is around 3KHz. Therefore, I boosted 3KHz a little to make the instrument as beautiful as possible.

2 IIR Delay

2.1 Introduction and Theory

The digital delay could be implemented with IIR (infinite impulse response) filter by setting the feedback delay line with a gain that is less than 1. Meanwhile, setting the number of delay samples m to an adequate constant makes the virtual reflections audible.[1]

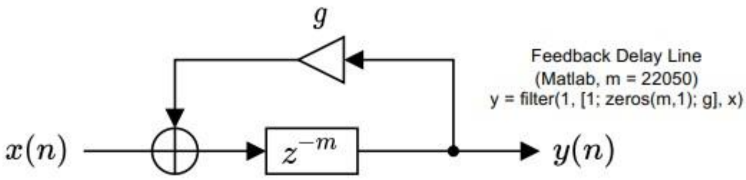

Figure 3: *Structure of a first-order IIR delay*

2.2code demo

```
m=1;
g=0.5;
b = [1;zeros(m,1);g];
y_f = filter(b,1,signal);
player_f = audioplayer(y_f,fs);
[H_f,w] = freqz(b,1);
plot(w/pi,20*log10(abs(H_f)))
xlabel('Normalized Frequency (\times\pi rad/sample)')
ylabel('Magnitude (dB)')
play(player_f)
```

2.3Changing Parameters

The reverberation in the feedback system sounds better than the feedforward system, but it would be annoying if the M is not big enough because when the delay factor is small, the filter becomes a comb filter and boosts hugely in specific high frequencies. Also, when the gain factor G is closer to 1, the more echoic (the echo generates faster) it sounds, and vice versa. If the G is more than 1, the disaster will happen to human hearing, as it becomes an unstable IIR[4].

3 Flanger

3.1Introduction

One of the easiest ways of building a digital flanger plugin is to use two forward delay paths. One of them is with a fixed delay length, and the other delay length is modulated with LFO (low-frequency oscillator). The LFO controls how wobbling the flanger effect is. This technique is also used for vocal vibrato modeling.

To make the effect better, the delay length *l* should better be half of the maximal value of the modulated delay length *m*.

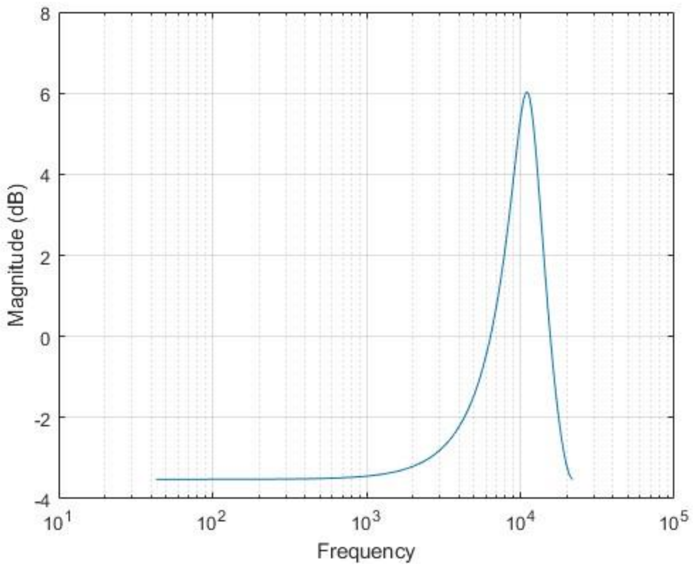

Figure 4: *Frequency response*

3.2 code demo

```
%% Modulation
A = 100;
rate = 0.5;
manual = 50;
period = 1/fs;
%% Flanger

for t = 1:length(signal)-A-manual
    y(t) = (signal(t+(manual+A)/2) + signal(t +(manual + ...
        round(A*sin(2*pi*t*rate*period))))/2);
end
sound(y,44100);
```

As to the simple definition of the feed forward flanger's structure, manually implement LFO with a sinusoid, and also, the static path is added. It sounds nice, like the '60s The Doors and Jimi Hendrix's timbre, in which you can hear a slow pitch shifting.

From the chart, we can see that when gain factor b0 of the modulation path reaches 1, the flanger effect would be the most obvious[2].

4 Chorus

The chorus plug-in basically refers to the same source with random start points and gain factors. These make the output sounds like an actual chorus but not simply multiplied like FIR or IIR delay.

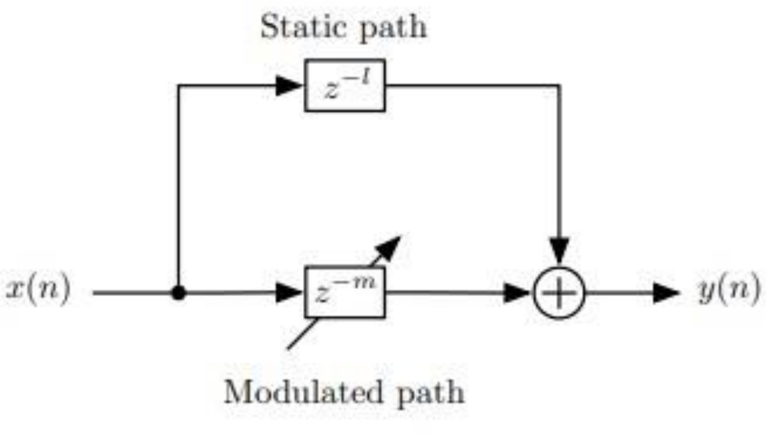

Figure 5: *Basic structure of flanger effect*

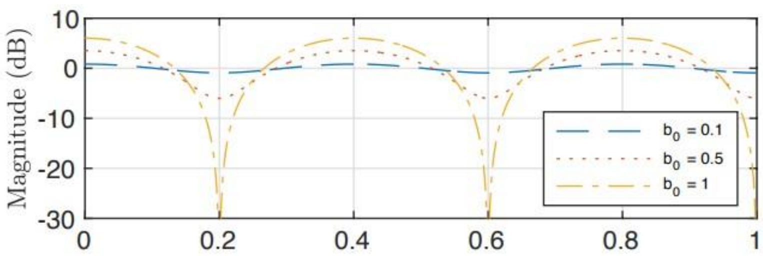

Figure 6: *Basic structure of flanger effect*

The digital guitar chorus plug-in can be modeled as multiple feed forward paths with modulated delay lines. The modulation wave forms may be sine waves or lowpass-filtered noise[3].

4.1code demo

```
%% Chorus
for t = 1:length(signal)-A-manual    y(t) = signal(t) + ...
                    (sin(2*pi*t*rate*period)+cos(2*pi*t*rate*period)).*(signal(t
            +(manual + round(A*sin(2*pi*t*rate*period))))/2);
end
sound(y,44100);
%As to the block diagram given from the slide, here implements two %feedforward path with
the modulated gained by sinewave and ...
    cosine wave.
...
```

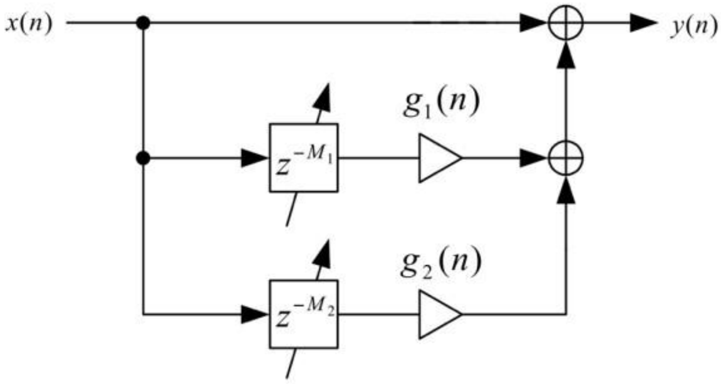

Figure 7: *Basic structure of chorus effect*

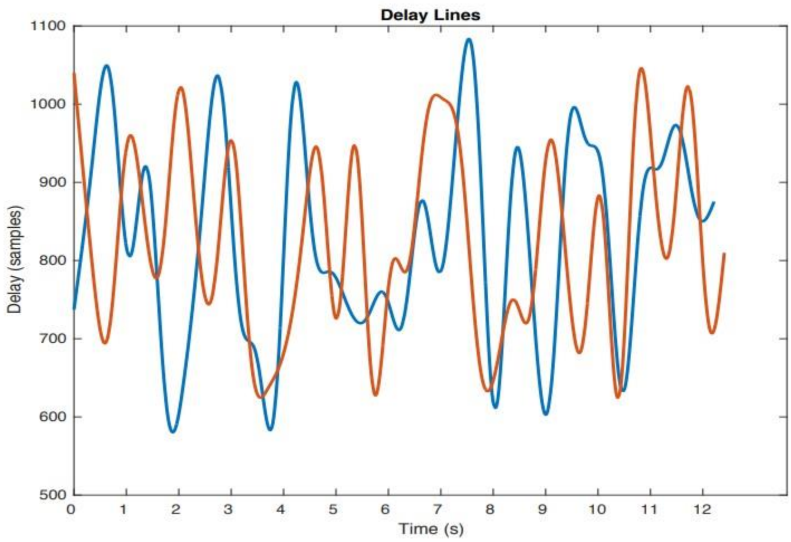

Figure 8: *2nd-order chorus delay*

5 Conclusion

Shelving filters can be used together as an Equalizer so that we can filter out the sounds we want. IIR delay can be used to achieve the effect of sound repeating while distancing, which I usually use it with vocals. Flanger and chorus can be used to achieve wider variation of dynamics.

References

[1] Chowdhury, J. A comparison of virtual analog modelling techniques for desktop and embedded implementations. *arXiv preprint arXiv:2009.02833* (2020).

[2] Pakarinen, J., Valim¨ aki, V., Fontana, F., Lazzarini, V., and Abel,¨ J. S. Recent advances in real-time musical effects, synthesis, and virtual analog models. *EURASIP Journal on Advances in Signal Processing 2011* (2011), 1–15. [3] Valim¨ aki, V.¨ Virtual analog audio signal processing.

[4] Valim¨ aki, V., Pakarinen, J., Erkut, C., and Karjalainen, M.¨ Discrete-time modelling of musical instruments. *Reports on progress in physics 69*, 1 (2005), 1.